**第一章 概述**

**简述网络安全的概念：**网络安全是指网络系统的硬件、软件及系统中的数据受到保护，不因偶然的或者恶意的原因而遭受到破坏、更改、泄漏，系统连续可靠正常的运行，网络服务不中断。网络安全从其本质上讲就是网络上的信息安全。

网络安全大体上可以分为信息系统（如主机、网络服务器）的安全、网络边界的安全及网络通信的安全。

机密性、完整性、可用性也称为信息安全的三要素。

**机密性**：是指保证信息不能被非授权访问，即使非授权用户得到信息也无法知晓信息内容，因而不能使用。它的任务时确保信息不会被未授权的用户访问，通常通过访问控制阻止非授权用户获得机密信息，通过加密变换阻止非授权用户获知信息内容。

**完整性**：是指维护信息的一致性，即信息在生成、传输、存储和使用过程中不应发生人为或非人为的非授权篡改。一般通过访问控制阻止篡改行为，同时通过消息摘要算法来检验信息是否被篡改。信息的完整性包括两个方面： (1) 数据完整性：数据没有被(未授权)篡改或者损坏； (2) 系统完整性：系统未被非法操纵，按既定的目标运行。

**可用性**：是指保障信息资源随时可提供服务的能力特性，即授权用户根据需要可以随时访问所需信息。可用性是信息资源服务功能和性能可靠性的度量，涉及到物理、网络、系统、数据、应用和用户等多方面的因素，是对信息网络 总体可靠性的要求。

还有不可抵赖性、真实性、可控性、可审查性和保鲜性。

**网络安全体系结构**是安全服务、安全机制、安全策略及相关技术的集合。①**安全服务**：X.800对安全服务做出定义：为了保证系统或数据传输有足够的安全性，开放系统通信协议所提供的服务. RFC2828也对安全服务做出了更加明确的定义：安全服务是一种由系统提供的对资源进行特殊保护的进程或通信服务。②**安全机制**：安全机制是一种措施或技术，一些软件或实施一个或更多安全服务的过程。常用的安全机制有认证机制、访问控制机制、加密机制、数据完整性机制、审计机制等。③**安全策略**：所谓安全策略，是指在某个安全域内，施加给所有与安全相关活动的一套规则。所谓安全域，通常是指属于某个组织机构的一系列处理进程和通信资源。安全策略（一套安全规则）由该安全域中所设立的安全权威机构制定，并由安全控制机构来描述、实施或实现。④**安全技术：**安全技术是与安全服务和安全机制对应的一序列算法、方法或方案，体 现在相应的软件或管理规范等之中。

**分层的网络安全体系结构**看作是网络协议层次、安全功能和安全技术的集合。

**网络攻击和网络保护：**

**网络攻击**：指利用目标系统的安全缺陷，为窃取、修改、伪造或破坏信息，以及降低、破坏网络使用效能而采取的各种措施和行动。

**网络防护**：指为保护己方网络和设备正常工作、信息数据安全而采取的措施和行动。

**网络攻击的目标**是破坏网络信息系统的安全行（机密性、完整性、可用性）

**网络防护的目标**是保证己方网络数据的机密性、完整性、可用性、真实性、可控制性与可审查性。

**TCP/IP网络体系结构及计算机网络的脆弱性**

①网络基础协议存在漏洞②网络硬件存在安全隐患③软件缺陷和安全漏洞④操作系统存在安全隐患⑤网络体系结构的安全风险

**举例说明计算机网络面临的主要威胁**

**安全威胁**主要利用网络与信息系统存在的脆弱性和网络管理中的漏洞。 **网络安全面临的主要威胁：**

①各种自然因素 ②内部窃密和破坏③信息的截获和重演④非法访问⑤破坏信息的完整性⑥欺骗⑦抵赖⑧破坏系统的可用性

**计算机网络安全的主要技术与分类**：以系统的角度可分为：①网络侦查②网络攻击③网络防护；以网络安全的主要技术可分为①网络侦查技术②网络攻击技术③网络防护技术。

**简述网络侦查所涉及的主要技术**

网络侦察是指运用各种技术手段、采用适当的策略对目标网络进行探测扫描，获得有关目标计算机网络系统的拓扑结构、通信体制、加密方式、网络协议与操作系统、系统功能，以及目标地理位置等各方面的有用信息，并进一步判别其主控节点和脆弱节点，为实施网络攻击提供可靠的情报保障。①端口探测技术（开源软件nmap） ②漏洞探测技术(模拟攻击、信息型漏洞探测) ③隐蔽侦察技术 ④渗透侦察技术

**简述网络攻击所涉及的主要技术和手段**

计算机网络攻击是指利用目标计算机网络系统的安全缺陷（漏洞），为窃取、修改、伪造或破坏信息，以及降低、破坏网络使 用效能而采取的各种措施和行动。其目的是破坏网络信息系统的安全属性。

**①拒绝服务（DoS）**指攻击者通过向目标主机建立大量的连接请求，阻塞通信信道，延缓网络传输，挤占目标机器的服务缓冲区，以至目标计算机忙于应付，响应迟钝，直至网络瘫痪系统关闭 ②**入侵攻击**攻击者利用操作系统中内在缺陷或者对方使用的程序语言本身所具有的安全隐患等，非法进入本地或远程主机系统，获得一定的权限，进而可以窃取信息，删除文件，埋设后门等行为 ③病毒攻击 ④有害代码攻击 ⑤邮件攻击 ⑥诱饵攻击

**计算机病毒**就是一种典型的恶意代码，此外，还包括木马、后门、逻辑炸弹、蠕虫等。

**如何网络安全防护主要目标**

①“进不来”：使用访问控制机制，允许授权用户访问，阻止非授权用户进入网络，保证网络的可控性和可用性 ②“拿不走”：使用授权机制，实现对用户的权限控制，同时结合内容审计机制，实现对网络资源及信息的可控性 ③“看不懂”：使用加密机制，确保信息不暴露给未授权的实体或进程，实现信息的保密性 ④“改不了”：使用数据完整性鉴别机制，保证只有得到允许的人才能修改数据，确保信息的真实性和完整性 ⑤“走不脱”：使用审计，监控，防抵赖等安全机制，使得破坏者走不掉。并进一步对网络出现的安全问题提供调查依据和手段，实现信息的可审查性

**简述网络防护所涉及的主要技术和手段**

**①防火墙技术** 防火墙是实现网络访问控制的装置，是最基本的网络防护措施， 也是目前使用最广泛的一种网络安全防护技术。防火墙技术主要有两种：数据包过滤技术和代理服务技术。（sudo ufw status）②**入侵检测技术** 入侵检测是一种动态安全技术，通过对入侵行为的过程与特征的研究，使安全系统对入侵事件和入侵过程能做出实时响应。有两种主要的入侵检测技术：基于特征的检测和基于行为的检测， 也称为误用检测和异常检测。入侵检测系统从实现方式上一般分为两种，即基于主机的入侵检 测系统和基于网络的入侵检测系统。 ③**计算机病毒及恶意代码防治技术** 检测病毒的主要方法是特征码及行为分析法，特征码是某种病毒或恶意代码的唯一特征，行为分析法通过判断代码是否有破坏信息系统的行为。 ④**密码技术** 密码技术主要研究数据的加密和解密。单密钥体制也称为传统密码体制，加密密钥和解密密钥相同；双密钥体制也称为公开密钥加密体制，主要用于密钥分配和数字签名⑤**认证技术** 认证主要包括身份认证和信息认证⑥**“蜜罐”技术**“蜜罐”是试图将攻击者从关键系统引诱开的诱骗系统。

1980中期发布了可信计算机系统安全评估准则**TCSEC**，如何从整体上采取积极的防护措施，加紧确立和建设信息安全保障体系，是世界各国正在研究的热点问题。

**黑客行为道德规范**

①不随便进行攻击行为②公开自己的作品③帮助其他黑客④义务地做一些力所能及的事情

**简述黑客精神/简述原始黑客精神**

①自由共享的精神，黑客文化的精髓 ②探索与创新的精神 ③合作的精神

**13.网络对抗主要涉及到哪些方面的技术**

网络侦查技术、网络攻击技术和网络防护技术

**第二章 网安基础知识**

**1.写出用net启动和关闭C盘共享、映射远程主机192.168.11.200\C$为本地磁盘Z：的命令**

net share C=C:\ （启动C盘共享）

net share C：\ /del (关闭C盘共享)

net use Z：\\192.168.11.200\C$

其余net命令：（最后一个名字看情况赋值）

**启动关闭服务**：net start servuedname 和net stop servicecname

**启动关闭共享**：net share sharename 和 net share sharename /del

**影射磁盘和删除映射磁盘**：net use drivename \\ip\drive 和net use drivename /del

**添加删除用户**：net user username password /add 或 /del

**激活和关闭guest帐号**：net user guest/active:yes和net user guest /active:no

**2.远程登录使用telnet可以从客户端登录服务器**

telnet IP [port]

**3.文件传输命令ftp**

ftp [-v] [-d] [-i] [-n] [-g] [-s:filename] [-a] [-w:windowsize] [-A] [host]

**4.添加计划任务命令schtasks.exe**

使用schtasks命令可安排在特定日期和时间运行指定程序：SCHTASKS /Create /S workstation /SC DAILY /ST 10:36 /TN report01 /TR "C:\Windows\notepad.exe"

**5. 查看修改文件夹权限命令cacls**

cacls filename [/T][/E][/C] [/G user : perm] [/R user […]] [/P user : perm […]] [/D user […]]

**6. 回显命令echo**

使用echo命令可以在屏幕上显示指定的信息，利用echo和>>(添加末尾)或>(覆盖)符号可以把命令结果导出到某文件中。

**7. 命令行下的注册表操作**

REG Operation [参数列表]

**8. 查看当前系统用户情况命令query**

QUERY { PROCESS | SESSION | TERMSERVER | USER }

**9. 终止会话命令logoff**

Logoff [sessionname|sessionid] [server:servername] [/V]

**10. 物理网络查看命令ping**

Ping IP (可看TTL)

**11. 网络配置查看命令ipconfig**

Ipconfig /all或/renew

**12. 查看通信路由命令tracert**

诊断实用程序将包含不同生存时间(TTL）值的Internet控制消息 协议(ICMP)回显数据包发送到目标，以决定到达目标采用的路由。

**13. DNS查看nslookup**

**14. netstat显示协议统计和当前的TCP/IP网络连接。**

**15. route操作网络路由表**

**网络端口：**逻辑意义上的端口，一般是指TCP/IP协议中的端口，即网络协议 (网络)端口。端口号的范围从0~65535（比如用于浏览网页服务的80端口，用于FTP服务的21端口）等。1. **端口的作用**：与进程关联的一种数据结构 2. **端口的分类**：知名端口、动态端口；协议端口 3. **端口在入侵中的作用**：入侵的门窗 4. **端口的相关工具**：netstat和nmap 5. **端口的保护**：查看、判断、关闭

**服务：**进程是指在系统中正在运行的一个应用程序，程序是构成进程的组成部分之一。程序是静态的，而进程是动态的。从操作系统角度来看进程分为用户进程和系统进程。

**第三章 密码学基础**

**1.密码学概述**

信息安全的主要目标是保护信息的机密性、完整性和可用性，机密性主要通过密码技术实现。密码学是研究如何隐密地传递信息的学科，其首要目的是隐藏信息的涵义。密码学涉及信息的加密/解密及密码技术在信息传递过程中的应用。 • 早期的密码技术的安全性基于密码算法的保密，现代的密码技术要求密码算法公开、密钥必须保密，密码算法的强度基于计算的复杂性。

密码学相关学科：密码学、密码编码学和密码分析学。

**2.密码学需要提供三方面的功能**：鉴别、完整性和抗抵赖性。 • **鉴别**：消息的接收者应该能够确认消息的来源；入侵者不可能伪装成他人。 • **完整性**：消息的接收者应该能够验证在传送过程中的消息没有被修改；入侵者不可能用假消息代替合法消息。 • **抗抵赖性**：发送消息者事后不可能虚假地否认他发送的消息。

**3.简述对称加密算法的基本原理**

• 明文用M (Message，消息) 或P (Plaintext，明文)表示，它可能是比特流、文本文件、位图、数字化的语音流或者数字化的视频图像等。 • 密文用C (Cipher) 表示，也是二进制数据，有时和M一样大，有时稍大。通 过压缩和加密的结合，C有可能比P小些。 • 加密函数E作用于M得到密文C，用数学公式表示为： E(M)=C • 解密函数D作用于C产生M，用数学公式表示为： D(C)=M • 先加密、再解密，原始的明文将恢复出来，下式必须成立： D( E(M) )=M

基于密钥的算法通常有两类：对称密码算法和公开密钥算法（非对称算法）。对称密码算法有时又叫传统密码算法或单密钥密码算法，加密秘钥能够从解密密钥中推算出来，反之也成立。对称密码的安全性依赖于密钥的保密。

**4.对称密码算法的两个分支**

分组密码（块密码）算法（DES、DES3、IDEA、AES等）和序列密码（流密码）算法（RC4、A5、SEAL、PIKE）

**5.简述公开密钥算法的基本原理**

公开密钥算法（非对称算法）的加密密钥和解密密钥不同，而且解密密钥不能根据加密密钥计算出来，或者至少在可以计算的时间内不能计算出来。

加密密钥叫做公开密钥（公钥），解密密钥叫做私人密钥（私钥）。公开密钥K1加密表示为：Ek1(M)=C 私钥K2 解密为Dk2(C)=M

**安全协议：**密码协议：也称作安全协议，是使用密码学的协议，是以密码学 为基础的消息交换协议，其目的是在网络环境中提供各种安全服务。

**密码编码原则**：密码算法应建立在算法的公开不影响明文和密钥 的安全。

**6.DES算法的原理**

• 第一步：变换(置换)明文。对给定的64位比特的明文x，首先通过一个置换IP表来重新排列x，从而构造出 64位比特的x0，x0=IP(x)=L0R0，其中L0表示x0的前32比特，R0表示x0的后32位。

• 第二步：按照规则进行16轮迭代。规则为 Li = Ri-1 Ri = Li⊕f(Ri-1 ,Ki ) （i=1,2,3…16） 经过第一步变换已经得到L0和R0的值，其中符号⊕表示的数学运算是异或，f表示 一种置换，由S盒置换构成，Ki是一些由密钥编排函数产生的比特块。f和Ki将在后面介绍。

• 第三步：对L16R16利用IP-1作逆置换，就得到了密文y.

**7.RSA算法描述**

• **密钥计算方法**： 选择两个大素数p和q (典型值为1024位) 计算 n=p×q 和 z=(p-1) × (q-1) 选择一个与 z 互质的数，令其为 d 找到一个 e 使满足 e×d =1 (mod z) 公开密钥为 (e, n)，私有密钥为 (d, n)

• **加密方法**： 将明文看成比特串，将明文划分成k位的块P即可，这里k是满足2^k<n的最大整数。对每个数据块P，计算C＝P^e (mod n)，C即为P的密文。

• **解密方法**： 对每个密文块C，计算P＝C^d (mod n)，P即为明文。

• **Example**： 取p＝3，q＝11 则有n＝p×q=33，z＝(p-1)×(q-1) =(3-1) ×(11-1)=20 7和20没有公因子，可取d＝7 解方程7×e＝1(mod 20)，得到e＝3 公钥为（3, 33），私钥为（7, 33） • 加密： 若明文P＝4，则密文C＝P e (mod n)＝4 3 (mod 33)＝31 • 解密： 计算P＝Cd (mod n)＝317 (mod 33)＝4，恢复出原文

**RSA算法的安全性：**没有人找到一个多项式时间的算法来分解一个大的整数的因子，同时也还没有人能够证明这种算法不存在。

**8.消息摘要**

消息摘要的目的是将消息鉴别与数据保密分开，其基本设想是：发送者用明文发送消息，并在消息后面附上一个标签，允许接收者利用这个标签来鉴别消息的真伪。用于鉴别消息的标签需满足两个条件：能够验证消息的完整性，即能辨别消息是否被修改；标签不可能被伪造。目前使用比较多的是MD5和SHA。

为了辨别消息是否被修改，可以将一个散列函数作用到一个任意长的消息m上，生成一个固定长度的散列值H(m)，这个散列值称为该消息的数字指纹，也称消息摘要 (MD)。消 息的发送者对发送的消息计算一个消息摘要M1，和消息一起发给接收者；接收者对收到的消息也计算一个消息摘要M2，如果M2等于M1，则验证了消息的完整性，否则就证明了消息被篡改了。为了保证标签不可能被伪造，发送方可以用密码技术对消息摘要 M1进行加密保护，得到加密后的消息摘要C，接收方对C进行解密恢复M1，再与消息摘要M2比较，从而判断消息的完整性。

**9.数字签名**

**数字签名(Digital Signature)**是指用户用自己的私钥对原始 数据的消息摘要进行加密所得的数据，即加密的摘要。

数字签名将信息发送人的身份与信息传送结合运来，可以保证信息在传输过程中的完整性，并提供信息发送者的身份认证，以防止信息发送者抵赖行为的发生。数字签名的具体工作过程为：（1）发送方使用单向散列函数对要发送的信息进行运算，生成信息摘要（2）发送方使用自己的私钥，利用非对称加密算法，对生成的信息摘要进行数字签名（3）发送方通过网络将信息本身和已进行数字签名的信息摘要发送给接收方（4）接收方使用与发送方相同的单向散列函数，对收到的信息进行运算，重新生成信息摘要（5）接收方使用发送方的公钥对接收的信息摘要解密（6）将解密的信息摘要与重新生成的信息摘要进行比较，以判断信息在发送过程中是否被篡改过。

**10.消息的保密传输**

为了对消息进行保密传输，通常将公钥密码技术和对称密码技术结合起来使用。 ① 在发送方A随机生成一个对称密码算法的密钥K。 ② 然后用K对消息加密得到密文C并生成密文的数字摘要M，接着 用A的私钥对M签名，将密文C、用B的公钥加密的签名及K发送给接收方B。 ③ 接收方B进行相反的操作，就可以实现消息的保密传输及完整性验证。

**Homework.用PGP加密某个文件，如果接收该加密文件的用户为1个，加密文件的大小为24kb，如果接收该加密文件的用户为10个，问加密文件的大小是原来的10倍（240kb）吗？为什么？**

**PGP算法：**PGP随机生成一个密钥，用IDEA算法对明文加密，然后用RSA算 法对密钥加密。收件人同样是用RSA解出随机密钥，再用IEDA解 出原文。

不是，因为加密文件24kb包括两个部分，发送者的公钥和文件摘要部分。当有10个接收的用户时，文件摘要部分不需要再次进行加密，是需要10用户的私钥。所以不需要原来的10倍。

**第四章 VPN技术**

**0.简述 VPN 组成和功能**

**VPN的功能**是将互联网虚拟成路由器，将物理位置分散的 局域网和主机虚拟成一个统一的虚拟企业网。

VPN 主要由 4 个关键部分组成：VPN 服务器、加密算法、认证系统和 VPN 协议。这 4 个部分保证了 VPN 提供以下功能。(1)加密数据：保证通过公网传输的信息不被第三方非法截取。(2)信息验证和身份识别：能够确保信息的完整性、机密性，并且能够验证远程站点的身份。(3)支持多种协议：能够支持 Internet 上使用的多种协议，特别是应用层协议，以及 IP、IPX、AppleTalk 等。(4)访问控制：不同用户有不同的访问权限。

**1.根据访问方式的不同，VPN分成哪几类？**

①**远程访问VPN**。移动用户访问VPN连接，由远程访问的客户机提出连接请求，VPN服务器提供对VPN服务器或整个网络资源的访问服务。在此连接中，链路上第一个数据包总是由远程访问客户机发出。远程访问客户机先向VPN服务器提供自己的身份，之后作为双向认证的第二步，VPN服务器也向客户机提供自己身份。

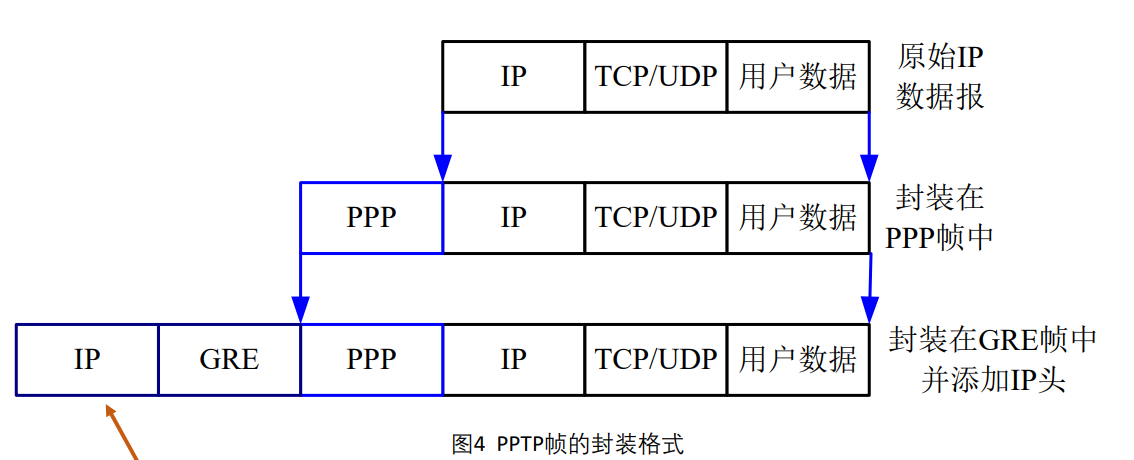
②**网关-网关VPN**。这种方案通过不安全的互联网实现两个或多个局域网的安全互联。在每个局域网的出口处设置VPN服务器，当局域网之间需要交换信息时，两个VPN服务器之间建立一条安全的隧道，保证其中的通信安全。

**2.根据隧道协议的不同，VPN分成哪几类？**

**隧道协议**（Tunneling Protocol）是一个网络协议的载体。使用隧 道的原因是在不兼容的网络上传输数据，或在不安全网络上提供一个安全路径。**具体可分类为**：第2层隧道协议(包含点到点隧道协议PPTP 、第二层转发协议L2F、第2层隧道协议L2TP，主要用于实现远程访问VPN)、第3层隧道协议(IPSec用于网络层实现数据包的安全封装，实现网关-网关VPN)、第4层隧道协议（SSL，在传输层上实现安全封装，主要用于保护两台主机的两个进程间的安全通信）、基于2和3层的隧道协议（利用MPLS路由器的标签特性实现的VPN）。

**3.PPTP VPN**

点对点隧道协议（PPTP） 是实现虚拟专用网（VPN）的方式之一。PPTP使用传输控制协议（TCP）创建控制通道来传送控制命令，以及利用通用路由封装（GRE）通道（数据通道）来封装点对点协议（PPP）数据包以传送数据。• PPTP以通用路由封装（GRE）协议向对方作一般的点对点传输， 通过TCP 1723端口来发起和管理GRE状态。



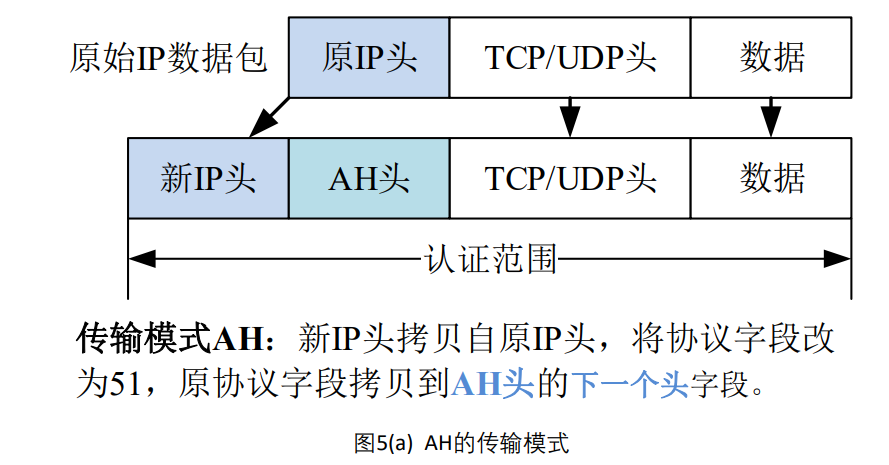
**4.IPSec**

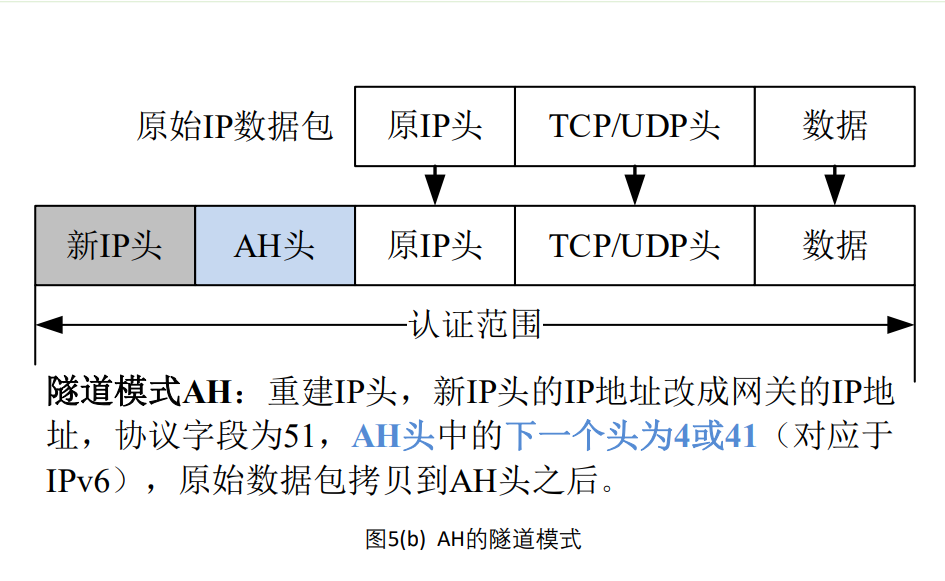
IPSec最重要的协议有三个：①认证头AH（AH为IP数据报实现无连接的完整性和数据源认证功能，并能抵抗重放攻击。）②封装安全有效载荷ESP（ESP实现保密性、数据源认证、无连接的完整性、抵抗重放攻击的服务(一种形 式的部分序列完整性)和有限的网络流的保密性。）③安全联盟SA（SA给出算法和数据的集合，以向AH或ESP的操作提供必须的参数。安全联盟和密钥管理协议ISAKMP提供了认证和密钥交换的框架。该框架支持手工配置的 预共享密钥以及通过其他方法获得的密钥，这些方法包括：Internet密钥交换 （IKE和IKEv2协议）、KINK、IPSEC KEY DNS记录）。

IPSec的两种工作模式：**传输模式**：传输模式用于两台主机之间的连接，在IP层封装主机—主机的分组。**隧道模式**：隧道模式用于两个网关之间的连接，在IP层封装网关—网关的分组，可穿过公共网络（如Internet）实现局域网之间的互联。AH和ESP均支持传输模式和隧道模式，实现认证和（或）加密等安全功能。

**认证协议AH**

**IP认证头AH**定义在RFC4302中，实现IP数据报的认证、完整性和抗重放攻击。AH数据报直接封装在IP数据报中，如果IP数据包的协议字段为51，表明IP头之后是一个AH头。





**封装安全载荷ESP**

IP封装安全载荷ESP （IP Encapsulating Security Payload）定义在 RFC 4303中，实现IP数据报的认证、完整性、抗重放攻击和加密。

**ESP**的数据报也直接封装在IP数据报中，如果IP数据包的协议字段为50，表明IP头之后是一个ESP数据报。 • ESP数据报由四部分组成，分别是：头部、加密数据（ESP载荷 +ESP尾）和ESP验证数据。

**安全关联和安全策略**

在AH和ESP头中有一个32bits的安全参数索引SPI，用于标识通信的两端采用的 IPSec 安全关联 SA(也称为安全联盟)。**安全关联**SA是单工的（即单向的），输出和输入都需要独立的SA。**一个安全关联由三个参数唯一决定**：安全参数索引号（一个与SA相关的位串，由AH和ESP携带， 使得接收方能选择合适的SA处理数据包）、IP目的地址（目前只允许使用单一地址，表示SA的目的地址）和安全协议标识（标识该SA是AH安全关联或ESP安全关联）。

**第五章 防火墙技术**

**1. 简述防火墙的定义：**防火墙是位于两个(或多个)网络之间执行访问控制的软件和硬件系统，它根据访问控制规则对进出网络的数据流进行过滤。

为了抵挡内部和外部网络的各种风险，可以在内部网络与外部网络之间设置一道屏障，用于过滤和监视内外网络之间的数据流，这种屏障就是防火墙。防火墙用于实现企业的网络安全策略，实施网络访问控制规则。

**防火墙对数据流的处理方式：**①允许数据流通过②拒绝数据流通过，并通知发送方③丢弃数据流，不通知发送方。

**2. 防火墙有哪些功能？**

①访问控制功能：这是防火墙最基本和最重要的功能，通过禁止或允许特定用户访问特定资源，保护内部网络的资源和数据。②内容控制功能：防止非法用户进入内部网络，也能禁止内网用户 访问外网的不安全服务③日志功能：记录通过防火墙的信息内容和活动④对网络攻击的检测和告警：发生可疑动作时防火墙作适当的报警⑤集中管理功能：针对不同网络情况和安全需要，指定不同安全策略，在防火墙上集中实施，使用时可能需要改变安全策略

**3. 防火墙的分类**

**按使用范围分类**：分为个人防火墙和网络防火墙。

**按防火墙在网络协议栈中的过滤层次分类：**包过滤防火墙、电路级网关防火墙和应用级网关防火墙。 包过滤防火墙主要根据网络层的信息进行控制，电路级网关防火 墙主要根据传输层的协议信息进行过滤，应用级网关防火墙主要根据应用层协议的信息进行过滤。

**4. 包过滤防火墙**

包过滤防火墙也称为分组过滤防火墙。静态数据包过滤发生在网络层上。根据访问策略的实现机制不同，又分为静态包过滤和动态包过滤。

**静态包过滤防火墙**也称为数据包过滤器（防火墙）。数据包过滤器通过数据包的头部信息来判断是接受还是拒绝数据包，它并不查看数据包载荷中的应用数据。这种防火 墙检查流经它的每个数据包，根据数据包本身所带的信息决定它的去留，而不用参考其他数据包的内容。

**工作原理：**(1)接收每个到达的数据包。 (2)对数据包按序匹配过滤规则，对数据包的IP头和传输字段内容进行检查。如果数据包的头信息与一组规则匹配，则根据该规则确定是转发还是丢弃该数据包。 (3)如果没有规则与数据包头信息匹配，则对数据包施加默认规则。

**动态包过滤防火墙**也称为状态（检测）防火墙。状态防火墙会通过对流经的数据包的分析查找通信中的数据流，根据数据流的信息来帮助判断是否让数据包通行。数据流提供了数据包的上下文 。状态防火墙有时还会检测一些常用协议的应用数 据（虽然可以检测的数据量是有限的）， 通过这些数据来识别和跟踪相关的数据流。

**工作原理：**1.首先检测每一个有效连接的状态，并根据这些信息决定网络的 数据包是否能够通过防火墙。 2. 然后通过从协议栈低层截取数据包，并将当前数据包及其状态 信息和其前一时刻的数据包及其状态信息进行比较，从而得到 该数据包的控制信息。 3. 接下来，动态包过滤模块就开始截获、分析并处理所有试图通 过防火墙的数据包，以保证网络的高度安全和数据完整。4.再接下来检验IP 地址、端口以及其他需要的信息以便决定该通 信包是否满足安全策略。 5. 最后它还把会相关的状态和状态之间的关联信息存储到动态连 接表中以随时更新其中的数据。通过这些数据，动态包过滤模 块就可以观测到后继的通信信息。

动态包过滤技术能够通过检查应用程序信息以及连接信息，来判断某个端口是否需要临时打开。

**包过滤防火墙设置方法**：网络管理员首先根据企业的安全策略定义一组访问控制规则，然后防火墙在内存中建立一张与访问控制规则对应的访问控制列表。

**静态包过滤防火墙的优点**：静态包过滤防火墙仅检查当前的数据包，是否允许通过的判决仅依赖于当前数据包的内容，检查的内容包括如下几部分：①源IP 地址；②目的IP地址；③应用或协议号；④源端口号；⑤目的端口号。因此，对数据包的检测是孤立的、无状态的。

(1)对网络性能的影响较小； (2)成本较低； (3)对用户透明。

**静态包过滤防火墙的缺点：**(1) 安全性较低(2) 缺少状态感知能力 (3) 容易遭受IP欺骗攻击 (4) 创建访问控制规则比较困难。

**动态包过滤防火墙的优点：**（1）高安全性（2）高性能（3）伸缩性和易扩展性（4）针对性（5）应用范围广。**动态包过滤防火墙的缺点：**（1）没有对净荷部分过滤，因此只具有较低的安全性（2）容易遭受IP欺骗攻击（3）难以创建规则（4）连接建立时没有遵循RFC建议的三步握手协议可能会引入额外的风险，如果连接建立时仅使用两次握手，可能导致防火墙在Dos攻击时耗尽资源。

**5.应用代理防火墙**

应用级网关截获进出网络的数据包，对数据包的内容进行检查， 如果符合所制定的安全规则，则允许数据通过；否则根据安全策 略的要求进行处理。比如：可以直接丢弃数据包，也可以删除数据包的不良内容，将改变的数据包传递到通信的另一端。**应用级网关的优点**：（1）在已有安全模型中安全性较高（2）具有强大的认证功能（3）具有超强的日志功能（4）应用级网关防火墙的规则配置比较简单。**应用级网关的缺点：**（1）灵活性很差（2）配置繁琐（3）性能不高，有可能成为网络瓶颈。

**6堡垒主机**

堡垒主机是一种配置了较为全面的安全防范措施的 网络上的计算机，它为网络间的通信提供了一个阻塞点。**特征：**堡垒主机硬件运行较为安全的操作系统、网络管理员认为必要的服务才安装在堡垒主机上、当允许一个用户访问代理服务时，堡垒主机可能会要求进行额外认证、每一个代理都只能支持标准应用服务命令集中的一个子集、每一个代理只允许访问指定主机的通信，支持对通信进行详细的审计、代理之间相互独立等。**屏蔽主机模式防火墙**：堡垒主机担任了身份鉴别和代理服务的功能。这样的配置比单独使用包过滤防火墙或应用层防火墙更加安全。**缺陷**：一旦过滤路由器遭到破坏，堡垒主机就可能被越过，使得内部网络完全暴露。**屏蔽子网模式防火墙**：更安全，采用了两个包过滤器，一个位于堡垒主机和外部网络之间，另一个位于堡垒主机和内部网络之间。

**7防火墙的发展动态和趋势：**

**缺陷：**①防火墙不能防范不经由防火墙的攻击，有多种方法可以绕过防火墙，比如：内网用户vpn到外网。②防火墙目前还不能防止感染了病毒的软件或文件的传输。③防火墙不能防止数据驱动式攻击，比如：缓冲区溢出攻击。④防火墙还存在着安装、管理、配置复杂的缺点。

**第六章 入侵检测技术**

**入侵检测的概念及模型：**入侵就是试图破坏网络及信息系统机密性、完整性和可用性等安全属性的行为，入侵方式一般有：未授权的用户访问系统资源、已授权用户企图获得更高权限。所有能够执行入侵检测任务和实现入侵检测功能的系统都可称为入侵检测系统（IDS），系统包含数据收集器、检测器、知识库和控制器。

**1. IDS 的任务**

(1)信息收集：收集方式为①系统和网络的日志文件②目录和文件中的的异常改变③程序执行中的异常行为④网络活动信息。（2）信息分析：①模式匹配，将收集到的信息与已知的网络或系统入侵模式的特征数据库进行比较，从而发现违背安全策略的行为。②统计分析：创建系统现象的统计属性，与实际行为比较③完整性分析：完整性分析检测某个文件或对象是否被更改。（3）安全响应：分为主动响应和被动响应。**IDS提供的主要功能**：①网络流量的跟踪与分析功能②已知攻击特征的识别功能③异常行为的分析、统计与响应功能④特征库的在线和离线升级功能⑤数据文件的完整性检查功能⑥自定义的相应功能⑦系统漏洞的预报警功能⑧IDS探测器集中管理功能。**IDS的分类**：(1)基于网络的入侵检测系统(NIDS)，数据来源于网络上的数据流，NIDS能够截获网络中的数据包，提取 其特征并与知识库中已知的攻击签名相比较，从而达到检测目的。**优点**：检测速度快、隐蔽性好、不容易受到攻击、不消耗被保护主机的资源；**缺点**：有些攻击是从被保护的主机发出的，不经过网络，因而无法识别。(2)基于主机的入侵检测系统(HIDS)，数据来源于主机系统，通常是系统日志和审计记录。(3)分布式入侵检测系统(DIDS)。这种系统能够同时分析来自主机系统审计日志和网络数据流，一般为分布式结构，由多个部件组成。典型的DIDS 采用控制台/探测器结构，NIDS和HIDS作为探测器放置在网络的关键节点，并向中央控制台汇报情况。

**2．通用入侵检测框架CIDF模型及入侵检测原理**

**CIDF提出的主要目的：**①IDS构件共享②数据共享③完善互用性标准。CIDF模型包含事件产生器、事件分析器、相应单元、事件数据库和目录服务构件组成。**入侵检测系统的处理模式：**信息源、分析和相应。

**3. 什么是误用检测?基于误用检测原理的入侵检测方法有哪些?**

误用检测技术又称基于知识的检测技术。它假定所有入侵行为和手段(及其变种)都能够表达为一种模式或特征，并对已知的入侵行为和手段进行分析，提取检测特征，构建攻击模式或攻击签名，通过系统当前状态与攻击模式或攻击签名的匹配判断入侵行为。**误用检测技术的优点**在于可以准确地检测已知的入侵行为，**缺点**是不能检测未知的入侵行为。

**4. 什么是异常检测?基于异常检测原理的入侵检测方法有哪些?**

异常检测技术又称为基于行为的入侵检测技术，用来识别主机或网络中的异常行为。它假设攻击与正常的(合法的)活动有明显的差异。异常检测首先收集一段时间操作活动的历史数据，再建立代表主机、用户或网络连接的正常行为描述，然后收集事件数据并使用一些不同的方法来决定所检测到的事件活动是否偏离了正常行为模式，从而判断是否发生了入侵。**基于异常检测原理的入侵检测方法**有以下几种：(1)统计异常检测方法；(2)特征选择异常检测方法：(3)基于贝叶斯推理异常检测方法：(4)基于贝叶斯网络异常检测方法；(5)基于模式预测异常检测方法。

**Snort**：是一个基于libpcap的轻量级网络入侵检测系统。由解码器、预处理器、检测引擎、输出插件和日志构成。

**5. 分析 NIDS 的脆弱性**

**反NIDS的目标是**：使NIDS检测不到入侵行为的发生，或无法对入侵行为做出响应，或无法证明入侵行为的责任。

**脆弱性**：(1)检测的工作量大：NIDS 的检测是资源密集型的，这在某种程度上使 NIDS 更加容易遭受DoS 攻击。(2) 检测方法的局限性：复杂的、智能化方法的作用十分有限。特征匹配成为 NIDS 分析引擎的一个不可或缺的模块功能。特征匹配作为一种轻量级的检测方法有其固有的缺陷，缺乏弹性（尤其是字符串匹配），如何完备定义匹配特征（也即匹配特征库的完备性）是决定检测性能的一个关键问题。(3) 网络协议的多样性与复杂性：NIDS 须对大部分协议进行模拟分析检测工作，这会使得分析引擎变得臃肿而效率低下。(4) 系统实现的差异：NID 可能不能完全理解所保护主机对数据的解析，理解不一致导致漏报或误报。

**第七章 Windows及Linux系统的安全**

**1. 计算机系统的安全级别**

信息系统的安全等级从低到高划分为D、C、B和A四个等级，C和B 又分成多个子级。D：最小保护。C：自主保护（C1：自主安全保护，C2受控的访问保护）B：强制保护级（B1：标签式安全保护，B2：结构化保护，B3：安全域）A：经过验证的保护。从TCSEC的B2级到A1级，TCSEC要求所有对受信计算基（TCB） 的更改必须由配置管理进行控制。Windows NT的第一版带服务包3的NT3.5就取得了美国TCSEC(受信计算机系统评测标准)标准的C2安全级，一般认为Unix比Windows更安全，也达到了C2。**达到C2安全级的要求：**①安全登录机制②自主访问控制机制③安全审计机制④对象重用保护机制。

WindowsNT及后续版本额外满足了两项B安全级别要求：信任路径功能和信任机制管理。

**2．Windows系统的安全防护**

常用安全防护措施：①使用NTFS②防止穷举法猜测口令③使用高强度的密码④正确设置防火墙⑤路由和远程访问中的限制⑥系统安全策略⑦安装第三方安全软件，及时打上补丁⑧断开重要的工作主机与外部网络的连接

**3. Windows安全性概述**

Windows10执行的安全性工作有三类：①身份标识和访问控制②信息保护③防恶意软件

**4. Linux的安全机制**

Linux用户可以分为三类：root、普通用户和系统用户。Root对系统有完全的控制权，root用户ID为0；普通用户默认拥有主目录，Linux采用自主访问控制策略；系统用户没有主目录，且从不登录。用户的信息（加密的口令）放在/etc/passwd文件中。

**5. Linux的安全防护**

常用防护措施：①使用高强度的口令②用户登录超时注销③禁止访问重要文件④允许和禁止远程访问⑤禁止不必要的SUID程序⑥及时为系统的已知漏洞打上补丁⑦保证一些应用服务的安全

**6. 几种典型的Linux应用攻击**

①缓冲区溢出攻击 ②格式化字符串攻击 ③输入验证攻击

**第八章 32位Linux系统的缓冲区溢出**

**1. 缓冲区溢出攻击**

做实验需要先关闭地址随机化机制，sudo sysctl -w kernel.randomize\_va\_space=0

**缓冲区溢出错误**：当向缓冲区拷贝数据时，若数据的长度大于缓冲区的长度，则多出的数据将覆盖该缓冲区之外的（高地址）内存，从而覆盖了邻近的内存。

**Shellcode:**是能实现攻击者目的，对目标漏洞进行攻击的代码。

**Linux IA32进程映象**

低地址 高地址

.text .data .bss Heap 未使用 Stack 环境

**缓冲区溢出实例：**

#include #include char Lbuffer[] = "01234567890123456789========ABCD";// 32 bytes void foo() { char buff[16]; strcpy (buff, Lbuffer); }

int main(int argc, char \* argv[]) { foo(); return 0; }

(gdb) disas foo

Dump of assembler code for function foo: 0x0804840b<+0> : push %ebp …

0x0804841d<+18> : call 0x80482e0 <strcpy@plt> … 0x08048427<+28> : ret

End of assembler dump.

(gdb) b \*(foo+0) Breakpoint 1 at 0x804840b

(gdb) b \*(foo+18) Breakpoint 2 at 0x804841d

(gdb) b \*(foo+28) Breakpoint 3 at 0x8048427

(gdb) display/i $eip

(gdb) r

(gdb) x/x $esp 0xbfffeeac: 0x080484ee(表示esp中保存的值与地址上的值)

记录堆栈指针esp的值，在此以A标记：A=$esp = 0xbfffeeac

(gdb) c Breakpoint 2, 0x0804841d in foo ()

(gdb) x/x $esp 0xbfffee80: 0xbfffee90

(gdb) 0xbfffee84: 0x0804a040

(gdb) x/x 0x0804a040 0x804a040 <Lbuffer>: 0x33323130

Lbuffer的地址0x804a040 保存在地址为0xbfffee84的栈中， buff的首地址=0xbfffee90保存在地址为0xbfffee80的栈中。令B=buff的首地址，则buff的首地址与返回地址所在栈的距离=

A-B= 0xbfffeeac-0xbfffee90 =0x1c=28,如果改写Lbuffer的内容超过28字节，则发生缓冲区溢出，并改写返回地址

(gdb) c

(gdb) x/x $esp 0xbfffeeac: 0x44434241(等于ABCD)

Ret处会到0x44434241这个地址。

**第九章 Linux32shellcode技术**

要成功地利用缓冲区溢出漏洞，必须解决三个技术问题： (1)跳转地址放在攻击串的什么位置（偏移，offset）； (2)跳转地址的值（调试目标进程，确定(或猜测)目标缓冲区的 起始地址+偏移）；

(3)编写期望(能实现某些功能)的shellcode。

shellcode中存在字符\x00，而\x00是字符串结束标志。由于shellcode是要作为字符串拷贝到缓冲区中去的，在\x00之后的代码将丢弃。因此，shellcode中不能包 含\x00 。**解决方案**：①修改汇编代码，用别的汇编指令代替会出现机器码\x00的汇编指令②对shellcode进行编码，把编码程序和编码后的shellcode作为新的shellcode。

如果被攻击的目标缓冲区较小，不足以容纳shellcode，则将shellcode 放在被溢出缓冲区的后面；如果目标缓冲区较大，足以容纳shellcode，则将shellcode放在被溢出缓冲区中。

**第十章 Windows系统的缓冲区溢出攻击**

**Win32进程映像的特点：**

1)可执行代码fun1, fun2, main存放在内存块0x0040xxxx的低地址端， 且按照源代码中的顺序从低地址到高地址排列。 2)全局变量(x, y, z)也存放内存块0x0040xxxx的低地址端，位于可执 行代码之上(起始地址高于可执行代码的地址)。初始化的全局变量存放在低地址，而未初始化的全局变量位于高地址。3)局部变量位于堆栈的低地址区：字符串变量虽然先定义，但是其起始地址小于其他变量，最后进栈；其它变量从低地址到高地址依次逆序存放。 4)函数的入口参数的地址位于堆栈的高地址区，位于函数局部变量之上。

**第十一章 Windows32 shellcode技术**

**第十二章 格式化字符串和SQL注入漏洞攻击**

如果攻击者可以控制输入的字符串(含打印格式)，则有可能利用该漏洞执行shellcode，从而入侵目标系统。例如printf(user\_input); // 直接输出用户的输入，有漏洞

**如何看某个变量的值：**printf函数依次遍历格式化字符串(在此为"A is %d and is at %08x, B is %u and is at %08x.\n")中的字符，如果该字符不是格式化参数的首字符（由百分号%指定），则复制该字符，若遇到一个格式化参数，就采取相应的动作，用当前栈的内容替换该格式化参数(如果是%s，则拷贝相应的字符串)，并将栈指针esp增加4（相当于pop指令）。**重点：**若格式化参数个数>给定参数的个数，则printf会从栈的当前位置开始，依次向esp增大的方向获得数据并打印。

**常用的格式化字符有**： %s：打印地址对应的字符串； %n：对该printf()前面已输出的字符计数，将数值存入当前栈指针指向的栈单元存储的地址中； %m.nx：十六进制打印，宽度为m，精度为n，在m前加0处理为 左对齐； • 其中%s和%n读或写进程的堆栈存储的地址，若该地址是无效的内存地址，则将引发段错误从而使进程崩溃。 • 重点：抵抗格式化字符串攻击的最好方法是不允许用户修改格式串。

void formatstr\_vul() { char user\_input[1024]; unsigned long int\_input;

int A=0x3435,B=0x5657,C=0x7879; // Original values of A, B and C. printf("&A=0x%x\t&B=0x%x\tC=0x%x.\n",&A,&B,&C); printf("A=0x%x\tB=0x%x\tC=0x%x.\n",A,B,C); // getting an integer from user puts("Please enter a integer:");

scanf("%u", &int\_input); // getting a string from user

puts("Please enter a string:");

scanf("%s", user\_input); // Vulnerable place printf(user\_input);

puts(""); // New values of A, B and C. printf("New values\tA=0x%x\tB=0x%x\tC=0x%x.\n",A,B,C); }

结果如下：

Please enter a integer: 33 Please enter a string: 0x%08x.0x%08x.0x%08x.0x%08x.0x%08x.0x%08x.0x%08x.0x%08x.0x%08x.0x%08x. 0xbfffebac.0x00005657.0x00007879.0x00000004.0x00000004.0x00000174.0x00000021.0x000 03435.0x00005657.0x00007879. New values A=0x3435 B=0x5657 C=0x7879.(分别为int\_input,A,B,C)

观察这10个输出值可知：从栈顶开始的第7个(4字节)单元开始保存变量 int\_input，A、B、C值。我们可以从这10个(4字节)单元中找出无效地址（例如第二个单元的值），用%s或%n构造格式串使进程崩溃。**使进程崩溃的原理为**：设计包含%s或%n的格式化字符串，使其对应的栈地址无效，运行结果出现段错误(segmentation fault)，进程崩溃。

**如何读取任意内存的值**：从栈顶开始的第7个(4字节）单元开始保存变量int\_input, A, B, C的值，第7个(4字节）单元int\_input是用户输入、是可以控制的。如果想读取某个内存单元的值，可以将int\_input设置为内存地址，然后设置第7个格式化参数为%s,就可以打印出内存地址的值。

./v &A=0xbfffeba0 &B=0xbfffeba4 C=0xbfffeba8. A=0x3435 B=0x5657 C=0x7879.

Please enter a integer: 3221220264（C地址的十进制形式）

Please enter a string: 0x%08x.0x%08x.0x%08x.0x%08x.0x%08x.0x%08x.%s 0xbfffebac.0x00005657.0x00007879.0x00000004.0x00000004.0x00000174.yx

C=0x7879对应的字符串为"yx" 。

**改写指定内存地址单元的值：**原理：将目标地址放入堆栈之后，利用%m.n的格式，通过设定宽度和精度，控制%n的计数值，计数值就等于目标单元的值。

在此以修改A=0x6768为例，说明修改某个内存变量的步骤。将0x6768换算成十进制数为26472,说明%n得到计数值为26472， 即在%n之前共有26472个字符被打印。根据前面观察到的地址，在int\_input地址（第7个地址）之前，出现了5个32位的地址(0x%08x.)（这里是出现了5个无效地址，第一个地址是有效地址），每个地址对应**11个字符**；选择最后一个位置(第6个格式符的位置)采用%m.n的打印格式来增加字符； • 接下来是计算m和n的值，因为前面已经出现了5\*11=55个字符， 26472-55=26417。令n=26417，故最后一个（第6个格式符的位置） 位置写为%.26417u（或%.26417x等），其后跟%n。

• 输入的整数=A的地址= 0xbfffeba0 = 3221220256

• 输入的格式串为0x%08x.0x%08x.0x%08x.0x%08x.0x%08x.%.26417u%n。

**直接在格式串中指定内存地址：**

如果只允许输入字符串，即攻击者无法给int\_input赋值，或者改写的值太大而需要2次用%n改写，这时应该如何读写某个内存地址的值? **解决方案**：把要改写的内存地址写到格式串中。

void formatstr\_vul() { char user\_input[1024]; unsigned long int\_input; int A=0x3435,B=0x5657,C=0x7879; printf("&A=0x%x\t&B=0x%x\tC=0x%x.\n",&A,&B,&C); printf("A=0x%x\tB=0x%x\tC=0x%x.\n",A,B,C); puts("Please enter a string:");

scanf("%s", user\_input); printf(user\_input); puts(""); printf("New values\tA=0x%x\tB=0x%x\tC=0x%x.\n",A,B,C); }

**gdb调试程序**：

(gdb) disas formatstr\_vul

…

0x08048592<+167> : call 0x8048390 <printf@plt>…

(gdb) r

(gdb) x/x $esp 0xbfffeb40: 0xbfffeb5c(0xbfffeb40左边为栈顶， 0xbfffeb5c为user\_input首地址)

(gdb) p (0xbfffeb5c-0xbfffeb40)/4 $1 = 7

user\_input的首地址为0xbfffeb5c，位于栈顶开始的第7个(4字节）单元。

以下的运行结果也证明了这一点：

./v2 &A=0xbfffef80 &B=0xbfffef84 C=0xbfffef88. A=0x3435 B=0x5657 C=0x7879.

Please enter a string: ABCD%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x. ABCDbfffebac.00005657.00007879.00003435.00005657.00007879.44434241.78383025.

New values A=0x3435 B=0x5657 C=0x7879.

其中的第7个格式化输出0x44434241就是字符串"ABCD"的十六进制代码。将"ABCD"替换成要改写的内存地址，并且第7个格式化参数为 %n，正确设置第6个格式化参数，就可以改写内存的值。

**如何让scanf接受任意数字？**方法：重定向至写入文件中然后利用命令行的重定向功能，将该文件作为程序的输入。这样一来程序从文件中而不是从键盘中获得输入数据，就避开了任意数字的输入问题。

**具体实现**：先关闭地址随机化，假定要修改变量B的内容，设B的地址=0xbfffeb24=3221220132因此read2file从键盘输入整数3221220132和格式串

%08x.%08x.%08x.%08x.%08x.%08x.%08x.，运行结果如下：

./read2file

Please enter an address. 3221220132

Please enter the format string:

%08x.%08x.%08x.%08x.%08x.%08x.%08x. The string length is 39

./v2 < mystring

&A=0xbfffef80 &B=0xbfffeb24 C=0xbfffef88.

A=0x3435 B=0x5657 C=0x7879.

Please enter a string: ����bfffef8c.00005657.00007879.00003435.00005657.00007879. bfffeb24.

New values A=0x3435 B=0x5657 C=0x7879.

由以上的运行结果可知，已将目标地址0xbfffeb24放入了栈中，然后再采用与12.2.3中相同的方法，利用%n修改变量的值，即可完成攻击。

**如果变量的值太大，如B修改为0xfedcba98:**

如果变量的值太大，需要分两次写内存(每次写2字节)才能避免可能的段错误或进程的长时间运行。构造格式串，从命令行输入到文件mystring中

**Read2file2.c节选：**

\*address = u\_addr; 🡨数值小的2个字节

\*(address+1) = u\_addr+2;

\*(address+2) = u\_addr+2; 🡨数值大的2个字节

**执行./read2file2:**

Please enter an address. 3221220132

Please enter the format string: %08x.%08x.%08x.%08x.%08x.%.47711u%hn%.17476u%hn.%08x.%08x.

The string length is 70

其中地址计算为：

0xba98 - 5\*9 - 12 = 47711

0xfedc - 0xba98 = 17476

重定向后B的值变为0xfedcba98。

**如何将B的值变为0xba98fedc?** 交换Read2file2.c的第一行和第三行，然后同上计算即可。

**第十三章 64位系统的缓冲区溢出攻击**

与32位的系统相比，64位的溢出流程是类似的，返回地址为8个字节。

如果把地址看作字符串，则第7和第8字节为字符串结 束符'\0' ，即在构造攻击字符串时要考虑到跳转地址的最高2个字节为0（字符串结束符'\0'）。

对于64位系统，如果要成功利用strcpy导致的缓冲区溢出漏洞，则被攻击的缓冲区必须大到足以容纳shellcode